United Electronic Industries

The High-Performance Alternative

DNR/DNA-AI-211
Vibration Sensor Interface Boards
—
User Manual

24-bit, 4-channel, IEPE/ICP
2-wire Accelerometer Input Boards
for the PowerDNA and UEILogger Cubes
and the PowerDNR RACKtangle System

June 2009 Edition
PN Man-DNx-AI-211-0609

Version 1.3

© Copyright 1998-2009 United Electronic Industries, Inc. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, by photocopying, recording, or otherwise without prior written permission.

Information furnished in this manual is believed to be accurate and reliable. However, no responsibility is assumed for its use, or for any infringements of patents or other rights of third parties that may result from its use.

All product names listed are trademarks or trade names of their respective companies.

See UEI's website for complete terms and conditions of sale:

http://www.ueidaq.com/company/terms.aspx

Contacting United Electronic Industries

Mailing Address:

27 Renmar Avenue

Walpole, MA 02081

U.S.A.

For a list of our distributors and partners in the US and around the world, please see
http://www.ueidaq.com/partners/

Support:

Telephone:(508) 921-4600
Fax: (508) 668-2350

Also see the FAQs and online "Live Help" feature on our web site.

Internet Support:

Support support@ueidaq.com

Web-Site www.ueidaq.com

FTP Site ftp://ftp.ueidaq.com

Product Disclaimer:

WARNING!

DO NOT USE PRODUCTS SOLD BY UNITED ELECTRONIC INDUSTRIES, INC. AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS.

Products sold by United Electronic Industries, Inc. are not authorized for use as critical components in life support devices or systems. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness. Any attempt to purchase any United Electronic Industries, Inc. product for that purpose is null and void and United Electronic Industries Inc. accepts no liability whatsoever in contract, tort, or otherwise whether or not resulting from our or our employees' negligence or failure to detect an improper purchase.

Table of Contents

Chapter 1 Introduction 1

1.1 Organization 1

1.2 The DNx-AI-211 Analog Input Layer 3

1.3 Sensor Input Circuit 5

1.4 Device Architecture 5

1.5 Layer Connectors and Wiring 7
1.5.1 Antialiasing FIlters 7
1.5.2 Data Handling 8
1.5.3 LEDs 8
1.5.4 DNA-ACC-211 Breakout Board 8
1.5.5 Connectors 8

1.6 Jumper Settings for Board Position 9

Chapter 2 Programming with the High-Level API 11

2.1 Creating a Session 11

2.2 Configuring the Resource String 11

2.3 Configuring the Timing 12

2.4 Reading Data 12

2.5 Cleaning-up the Session 13

Chapter 3 Programming with the Low-Level API 14

Appendix A Accessories 24

A.1 DNA-ACC-211 Breakout Board 24
A.1.1 Mounting Multiple DNA-ACC-211 Boards 26

A.2 Other Accessories 26

A.3 Layer Calibration 27

Index 28

© Copyright 2009
United Electronic Industries, Inc.

Table of Figures

Chapter 1 Introduction 1
1-1 DNA-AI-211 Layer and DNA-ACC-211 Breakout Board 4
1-2 Typical Measuring Circuit for a Vibration Sensor 5
1-3 Block Diagram of DNA-AI-211 Channel Architecture 6
1-4 DB-37 I/O Connector Pinout 8
1-5 DNA-ACC-211 Connector Pinout 9
1-6 Jumper Block for DNA-AI-211 Board Position 10
1-7 Diagram of DNA-AI-211 Board Position Jumper Settings 10

Chapter 2 Programming with the High Level API 11
(None)

Chapter 3 Programming with the Low-Level API 14
(None)

Appendix A Accessories 24
A-1 Photo of DNA-ACC-211 Breakout Board 24
A-2 Mounting Multiple DNA-ACC-211 Breakout Boards 26

© Copyright 2009
United Electronic Industries, Inc.

Chapter 1 Introduction

This document outlines the feature set and use of the DNx-AI-211 vibration sensor interface layer(s) when used with the PowerDNA Core Module (DNA-AI-211) or the DNR-12-1G RACKtangle system (DNR-AI-211). This manual describes the following products:

- DNA-AI-211/DNR-AI-211, 24-bit, 4-channel, IEPE/ICP vibration sensor interface layer with 0-8 mA constant-current excitation and anti-aliasing filtering
- DNA-ACC-211 Breakout Board with 4 UNF connectors for connecting to 2-wire vibration sensors
- Accessory modules such as BNC cables.

1.1 Organization

This DNx-AI-211 User Manual is organized as follows:

- **Chapter 1 Introduction**
  This chapter provides an overview of DNx-AI-211 board/layer features, accessories, device architecture, connectivity, and logic.
- **Chapter 2 Programming with High-Level API**
  This chapter provides a general overview of procedures that show how to create a session, configure the session, and generate output on a DNx-AI-211 layer, working with the UeiDaq Framework High-Level API.
- **Chapter 3 Programming with the Low-Level API**
  This chapter describes the Low-Level API commands for configuring and using a DNx-AI-211 layer.
- **Appendix A - Accessories**
  This appendix provides a list of accessories available for use with a DNx-AI-211 layer.
- **Index**
  This is an alphabetical index of topics covered in this manual.

NOTE: A glossary of terms used with the UEI data acquisition systems and layers can be viewed and/or downloaded from www.ueidaq.com

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Manual Conventions

To help you get the most out of this manual and our products, please note that we use the following conventions:

Tips are designed to highlight quick ways to get the job done, or reveal good ideas you might not discover on your own.

NOTE: *Notes alert you to important information.*

CAUTION! *Caution advises you of precautions to take to avoid injury, data loss, and damage to your boards or a system crash.*

Text formatted in **bold** typeface generally represents text that should be entered verbatim. For instance, it can represent a command, as in the following example: “You can instruct users how to run setup using a command such as **setup.exe**.”

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

1.2 The DNx-AI-211 Analog Input Layer

This manual describes the DNx-AI-211 24-bit, 4-channel, IEPE/ICP Vibration Sensor Interface Board/Layer. It also describes the DNA-ACC-211 breakout board with four 10-32 UNF coaxial channel connectors and sensor status indicator LEDs. The technical specifications for the DNx-AI-211 Analog Input Layer are listed in **Table 1-1**.

Technical Specifications:

Channel Configurations	
Number of channels	4
Configuration	Two wire ICP/IEPE or Voltage
Sampling	Simultaneous (if sampled at identical freq.)
Isolation	350 VDC, (chan-to-chan and chan-to-chassis)
Input Specifications	
Resolution	24-bits
Signal to Noise Ratio	109 dB at max sample rate
Total Harmonic Distortion	-108 dB typical (at 1 kHz)
Sample Rate	1 SPS to 125 kSPS (500 kSPS per board)
Input Coupling	DC or AC (.1 Hz, 1.0 Hz or 10 Hz HP filter)
Input Ranges	+25/-13, ±12.5, ±5.0, ±2.5
Input Impedance	10 MOhm, minimum, 40 pF max
Offset Error	< 0.1 mV (5 µV per °C)
Gain Error	0.1% typ, <0.5% (5 ppm per °C)
Integral Nonlinearity (INL)	15 ppm, max
Anti-Aliasing Filtering (including both analog & digital filters)	
Filter Cutoff Frequency	0.49 times sample freq (3 db point)
Passband Ripple	±0.005 dB max
Stop Band Attenuation	-100 dB
Stop Band Frequency	0.547 times sample frequency
Bias Current Specifications	
Output Current Range	0.0 to 8 mA (software selectable)
Output Current Accuracy	± 1%
Short Circuit Protection	Continuous short will not cause damage
Maximum Output Voltage	25 VDC minimum
Dynamic Output Impedance	500 kOhm, minimum
Open/Short Detection	Automatic alarms for both high and low current at user selectable trigger points
Open/Short Annunciators	LEDs adjacent to the 10-32 UNF connectors
General Specifications	
Connections (1 per channel)	Standard 10-32 UNF coaxial connectors
ESD protection	15 kV
Operating temperature	tested -40 °C to +85 °C
Vibration *IEC 60068-2-6* *IEC 60068-2-64*	5 g, 10-500 Hz, sinusoidal 5 g (rms), 10-500 Hz, broad-band random
Shock *IEC 60068-2-27*	50 g, 3 ms half sine, 18 shocks @ 6 orientations 30 g, 11 ms half sine, 18 shocks @ 6 orientations
Humidity	0 to 95%, non-condensing
Altitude	0 to 70,000 feet
Power consumption	6.75 W, max (Cubes should include DNR-FAN option if mulitple AI-211s are installed)

Table 1-1. DNx-AI-211 Technical Specifications

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Figure 1-1 is a photo of the DNA-AI-211 Layer/board and its associated DNA-ACC-211 breakout board.

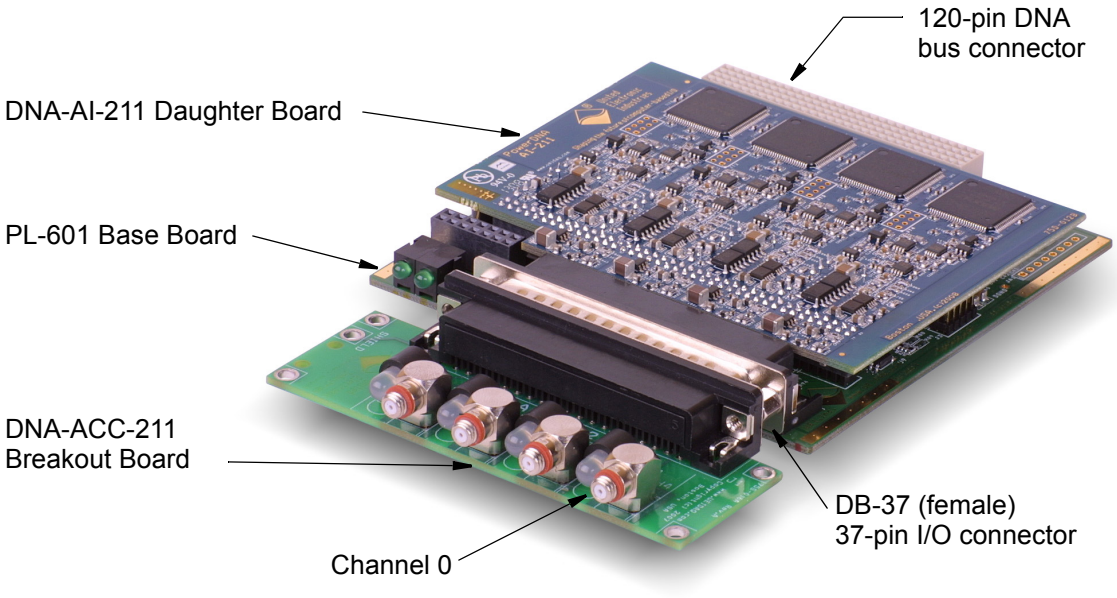


Figure 1-1. DNA-AI-211 Layer and DNA-ACC-211 Breakout Board

The DNR-AI-211 Layer is functionally identical to the DNA model, except for the bus connector — the DNR plugs directly into the backplane in a RACKtangle rack-mounted chassis instead of a PowerDNA Cube.

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

1.3 Sensor Input Circuit

The measurement signal from an IEPE or ICP vibration sensor is an AC waveform superimposed on a DC voltage. The DC voltage is produced by applying excitation from a constant current source to the sensor terminals so as to generate a voltage drop across the sensor. This voltage, which has a DC component (bias) from the constant current source and an AC component produced by the sensor, is then fed to the input of the AI-211 analog input board.

The AI-211 then filters the signal, removing the DC bias, if necessary, as described in the next section and passes it to an AD converter and then to an FPGA which filters it further.

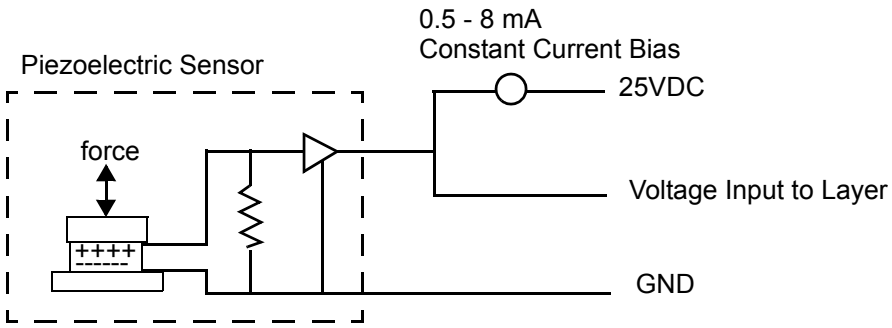


Figure 1-2. Typical Measuring Circuit for a Vibration Sensor

1.4 Device Architecture

The DNx-AI-211 Vibration Sensor Interface board has four individual channels, electrically isolated from each other and from the Cube. The product consists of three PC boards: (1) PL-601 base board, (2) an AI-211 daughter board plugged into the base board, and (3) a DNA-ACC-211 breakout board that plugs directly into the 37-pin I/O connector on the base board. The function of the breakout board is to provide four 10-32 UNF coaxial connectors and status indicator LEDs for the four vibration sensor inputs.

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

A Block Diagram of the board/layer is shown in **Figure 1-3**.

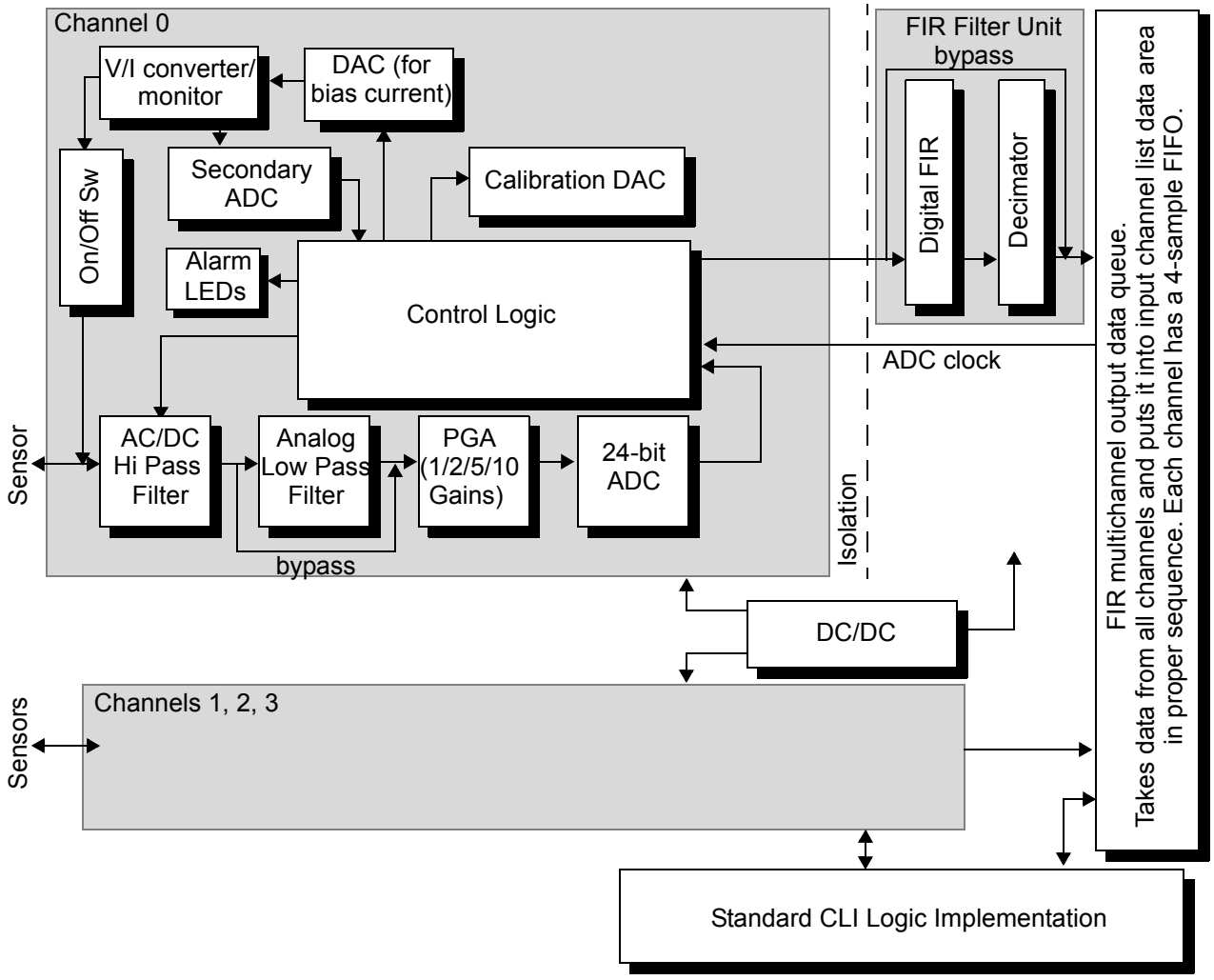


Figure 1-3. Block Diagram of DNA-AI-211 Channel Architecture

NOTE:

Each of the four AI-211 channels has independent power controls that are automatically engaged to reduce power consumption. If the channel list for an ACB (streaming) acquisition does not specify all four channels, any unused channels are automatically powered down for the duration of the acquisition session. At the end of the session, those unused channels are powered back up in case they may be required for the next acquisition session. When this re-powering occurs, a 100 millisecond delay is inserted into the setup time for the next acquisition to allow time for the power to stabilize. If the application requires that acquisition sessions be performed with less than 100 milliseconds between them, the channel list must specify all four channels, so that the power-down function is disabled and the subsequent power-up delay is eliminated.

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

1.5 Layer Connectors and Wiring

As shown in the Block Diagram of **Figure 1-3**, four constant-current sources provide excitation current for each of the four vibration sensor channels. This 0 to 8 mA excitation current, which can be user-set within ±1%, is used to bias the AC signal from the vibration sensor. The excitation current is produced by a user-programmable DAC that outputs a voltage to a V/I converter to generate a constant current. This excitation current is continuously monitored/controlled and can be switched on and off on a per-channel basis. The monitoring circuit uses a second 10-bit ADC that feeds back the reading to the control logic FPGA, which, in turn, sets the DAC.

The voltage from the secondary ADC is also used to turn status LEDs on/off as the excitation current changes value. The status LEDs associated with each channel are mounted on the DNA-ACC-211 breakout board, as described below.

When excitation is switched off, the channel can be used for sensing normal analog voltage inputs (same voltage ranges as with vibration sensors). The board is specifically designed, however, to connect to Industry-standard ICP and IEPE 2-wire piezoelectric vibration sensors when excitation is switched on. Each channel has its own 24-bit successive approximation A/D converter and all four channels can be sampled simultaneously at their maximum speed of 125 kSPS (500 kSPS total throughput). Use of the standard trigger/sync interface of the Cube or RACKtangle lets you synchronize other boards and other Cubes with this one for simultaneous sampling of all inputs.

Each of the four input channels is electrically isolated (transformer isolation) from the other channels and from the chassis.

The AC signal from the vibration sensor is biased by the DC excitation signal and then passed through a high-pass filter with programmable cutoff frequencies (0, 0.1, 1.0, or 10 Hz) and a 4-tap analog low-pass filter (48 kHz on or off), to attenuate the natural resonant frequency of the sensor and perform anti-aliasing filtering. It is then passed to the PGA and to the 24-bit SAR A/D Converter, which also provides automatic averaging over 8 readings.

1.5.1 Antialiasing FIlters

Since aliasing is a common problem with vibration or accelerometer measurements, UEI uses two types of anti-aliasing filters, digital and analog, to achieve the desired response. The 4-pole fixed analog filter placed ahead of the A/D Converter and set at 48 kHz removes aliasing errors from 50 kHz and up. This filter has a 24 dB/octave rolloff, which means that there are 3.3 octaves between 48 kHz and the 500 kHz non-aliasing Nyquist limit of the 1 MHz A/D converter. Therefore, the effect of any alasing error is reduced by at least 80 dB (3.3 x 24). Performance at lower sample rates is even better.

Automatic signal averaging (8-sample averaging) built into the A/D converter further reduces the effective noise

The digital filter in the FPGA is a programmable FIR filter that runs in real time. This gives a very sharp (“brick-wall”) filter with much steeper drop off than is possible with an analog filter and produces a uniform group delay (phase shift) between channels. It also means that no gain or offset error is introduced as might occur with an analog filter as you change filter frequencies.

The anti-aliasing filter achieves 109 dB attenuation within its stop band. Because each filter introduces an identical phase shift, no inter-channel phase jitter is present that could adversely affect data integrity.

The FIR filter is followed by a decimator to reduce the sampling rate and complete the anti-aliasing function.

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

1.5.2 Data Handling

Data output from the FIR filter unit is stored in sequence in the input channel list data area in 4-sample dedicated FIFOs, one for each channel. Refer to Chapters 2 and 3 for programming information.

1.5.3 LEDs

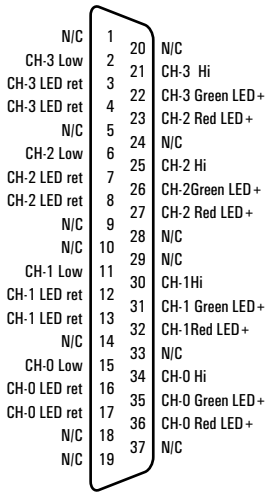
Continuous monitoring of the bias current in each channel enables you to detect an open, short, or off-normal condition in any channel. A two-color LED mounted on the ACC board next to each UNF connector indicates the current status of its associated measurement channel. Green indicates OK, red signifies an open circuit, and flashing red means a short circuit exists.

1.5.4 DNA-ACC-211 Breakout Board

The DNA-ACC-211 breakout board provides four standard 10-32 UNF coaxial connectors, one for each input circuit. The DNA-ACC-211 is designed for operation in harsh environments and has been tested for operation at 5g vibration, 50g shock, -40 to +85°C temperature, and altitude up to 70,000 feet. If you need a full-size BNC connector for your input, specify an accessory cable type DNA-CBL-BNC for each such input. Each cable is 2-feet long and has a 10-32 UNF coaxial connector on one end and a full size BNC connector on the other end.

1.5.5 Connectors

The pinout of the 37-pin connector for the DNx-AI-211 Layer board is shown in **Figure 1-4**. A physical layout of the board is shown in **Figure 1-1** on page 4.



Note: An LED+ pin brought high turns on the LED. For example, if Pin 35 is high, the channel 0 Green LED will be on (OK). If Pin 36 is brought high, the Channel 0 Red LED will be on. (If both are high, the LED will be orange. However, this is not a valid state on the AI-211. The valid states are green, solid red, flashing red.)

Figure 1-4. DB-37 I/O Connector Pinout

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Pinout of the DNA-ACC-211 breakout board is shown in **Figure 1-5**.

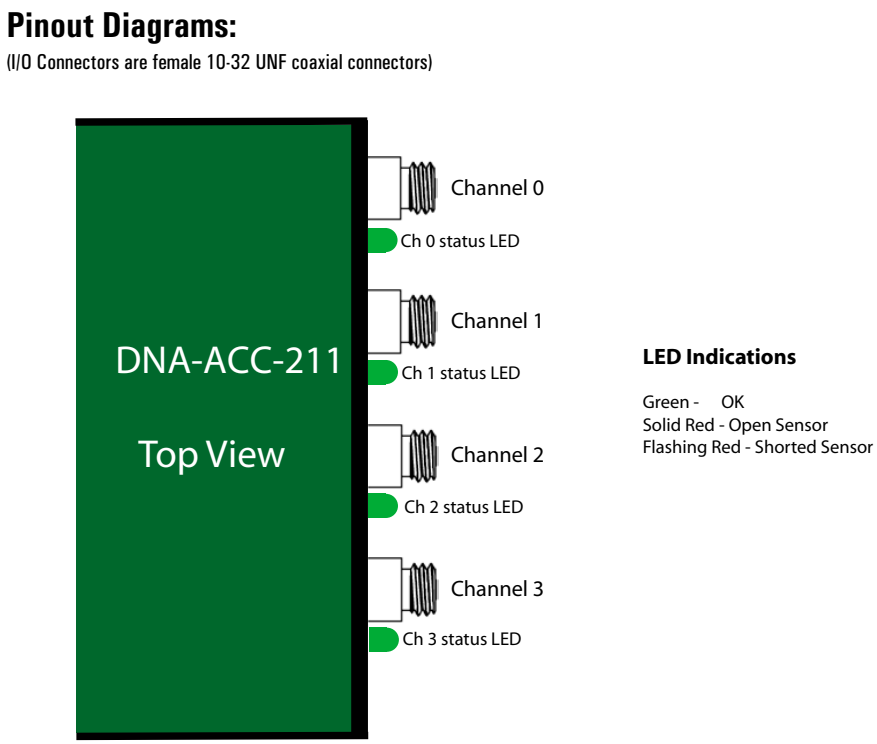


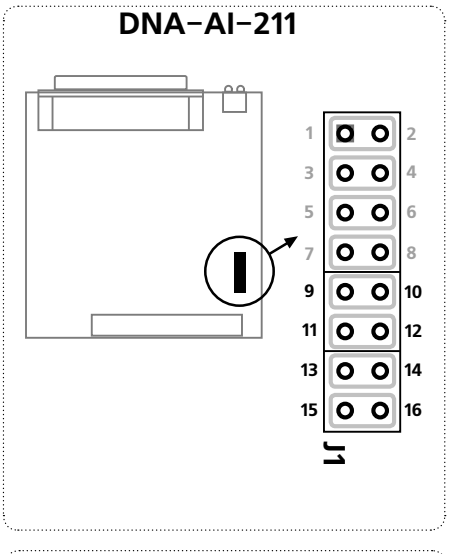
Figure 1-5. DNA-ACC-211 Connector Pinout

1.6 Jumper Settings for Board Position

Figure 1-6 shows the physical layout of DNA-AI-211 Base Board (DNA-PL-601), highlighted to show the 16-pin jumper block for setting the board position within the PowerDNA Cube.

NOTE: Board position jumpers are not provided with the DNR versions of the AI-211. The physical position of the board within the DNR RACKtangle™ enclosure is determined automatically by the system.

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.



See **Figure 1-7** for placement of jumpers for various board positions in a Cube.

Figure 1-6. Jumper Block for DNA-AI-211 Board Position

1.6.0.1 Jumper Settings

A diagram of the jumper block is shown in **Figure 1-7**. To set the board position jumpers, place jumpers as shown in **Figure 1-7**.

NOTE: Since all boards are assembled in Cubes before shipment to a customer, you should never have to change a jumper setting unless you change a board from one layer position to another in the field.

		Layer's Position as marked on the Faceplate*					
		I/O **1**	I/O **2**	I/O **3**	I/O **4**	I/O **5**	I/O **6**
Jx Pins	9-10	■	□	■	□	■	□
	11-12	■	■	□	□	■	■
	13-14	■	■	■	■	□	□
	15-16	■	■	■	■	■	■

* *All I/O Layers are sequentially enumerated from top to the bottom of the Cube*

□ - **Open** ■ - **Closed**

Figure 1-7. Diagram of DNA-AI-211 Board Position Jumper Settings

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Chapter 2 Programming with the High-Level API

This section describes how to program the DNx-AI-211 using the UeiDaq Framework API.

As UeiDaq Framework is object oriented, its objects can be manipulated in the same manner from different development environments such as Visual C++, Visual Basic, or LabVIEW.

The following section focuses on the C++ API, but the concept stays the same no matter what programming language you use.

Please refer to the “UeiDaq Framework User Manual” to get more information on using other programming languages.

2.1 Creating a Session

The Session object controls all operations on your device. Therefore, the first task is to create a session object:

```
CUeiSession session;
```

2.2 Configuring the Resource String

Framework uses resource strings to select which device, subsystem, and channels to use within a session. The resource string syntax is similar to a web URL:

```
<device class>://<IP address>/<Device Id>/
<Subsystem><Channel list>
```

For PowerDNA, the device class is **pdna**.

For example, the following resource string selects analog input channels 0,1 on device 1 at IP address 192.168.100.2:

"pdna://192.168.100.2/Dev1/Ai0,1"

Use the method **CreateAccelChannel()** to program the advanced features of the AI-211, such as the coupling, low-pass filter, sensor sensitivity, and excitation current.

The following call configures all analog input channels of a AI-211 set as device 1:

```
session.CreateAccelChannel("pdna://192.168.100.2/Dev1/
Ai0:3",
-100.0,// low limit
100.0,// high limit
24.0,// sensor sensitivity
5.0,// excitation current
UeiCouplingAC,// coupling
true);// low-pass filter
```

It configures the following parameters:

- **Low Limit:** the minimum expected measurement. The unit is the same as the sensor sensitivity unit. The low limit combined with the high limit determines which gain will be configured.
- **High Limit:** the maximum expected measurement. The unit is the same as the sensor sensitivity unit. The low limit combined with the high limit determines which gain will be configured

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

- **Sensor Sensitivity**: The sensor sensitivity. Its unit determines the unit of the measurements. For example, if the sensitivity is specified in mV/g, the measurements will be returned in number of g.
- **Excitation Current**: the excitation current used to power the IEPE sensor.
- **Coupling**: Configures the high-pass filter. DC coupling disables the filter, AC coupling enables a 0.1Hz high-pass filter.
- **Low pass Filter**: Enables or disables the low-pass analog filter. Cut off frequency is 48kHz.

The AI-211 is equipped with alarm circuitry that can notify you when the connection to a sensor is either open or shorted. It continuously measures the voltage on each channel and compares it to a low and high comparator values. When the value read is within the range, the LED on the terminal block is green. When the value is out of range, the LED changes to flashing red.

By default, a session is configured with the low comparator set to 1.0V and the high comparator set to 24V.

You can change the comparator values using the following methods on any of the channel object instances:

```
// Get channel 0 object
CueiAccelChannel* pChan =
dynamic_cast<CueiAccelChannel*>(session.GetChannel(0));

// Update comparator values
pChan->SetLowExcitationComparator(5.0);
pChan->SetHighExcitationComparator(15.0);
```

2.3 Configuring the Timing

You can configure the AI-211 to run in simple mode (point by point), buffered mode (ACB mode), or DMAP mode.

- In simple mode, the delay between samples is determined by software on the host computer.
- In buffered mode, the delay between samples is determined by the AI-211 on-board clock and data is transferred in blocks between PowerDNA and the host PC.
- In DMAP mode, the delay between samples is determined by the AI-211 on-board clock and data is transferred one scan at a time between PowerDNA and the host PC.

The following sample shows how to configure the simple mode. Please refer to the “UeiDaq Framework User Manual” to learn how to use the other timing modes.

```
session.ConfigureTimingForSimpleIO();
```

2.4 Reading Data

Reading analog data from the AI-211 is done using a reader object.

The following sample code shows how to create a scaled reader object and read samples.

```
// Create a reader and link it to the session's stream
```

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

```
CUeiAnalogScaledReader
aiReader(session.GetDataStream());

// Read measurements in Gs
double Gs[48];
aiReader.ReadSingleScan(Gs);
```

2.5 Cleaning-up the Session

The session object will clean itself up when it goes out of scope or when it is destroyed. However, you can manually clean up the session (to reuse the object with a different set of channels or parameters), as follows.

```
session.CleanUp();
```

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Chapter 3 Programming with the Low-Level API

This section illustrates how to program the PowerDNA cube using the Low-level API. The low-level API offers direct access to PowerDNA DAQBios protocol and also allows you to access device registers directly.

However, we recommend that, when possible, you use the UeiDaq Framework High-Level API (see Chapter 2), because it is easier to use.

You should need to use the low-level API only if you are using an operating system other than Windows.

The low-level functions for an AI-211 layer are described below.

DNA-AI-211 Layer Functions

Function: **DqAdv211Read**

Syntax: `int DqAdv211Read(int hd, int devn, int CLSize, uint32 *cl, uint16 *bData, double *fData)`

Command: `DQE`

Input:

`int hd`	Handle to the IOM received from DqOpenIOM()
`int devn`	Layer inside the IOM
`int CLSize`	Number of channels
`uint32 *cl`	Pointer to channel list
`uint32 *bData`	Pointer to raw data received from device
`double *fData`	Pointer to store converted voltage data (NULL if not required)

Output:

`uint32 *bData`	Raw data received from device
`double *fData`	Converted voltage data

Return:

`DQ_NO_MEMORY`	Error allocating buffer
`DQ_ILLEGAL_HANDLE`	Illegal IOM Descriptor or communication wasn't established
`DQ_BAD_DEVN`	Device indicated by devn does not exist or is not an AI-211
`DQ_BAD_PARAMETER`	CLSize is not between 1 and `DQ_MAXCLSIZE`, `bData` is NULL, or a channel number in `cl` is too high
`DQ_SEND_ERROR`	Unable to send the Command to IOM
`DQ_TIMEOUT_ERROR`	Nothing is heard from the IOM for timeout duration
`DQ_IOM_ERROR`	Error occurred at the IOM when performing this command
`DQ_SUCCESS`	Successful completion
`Other negative values`	Low level IOM error

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Description:

When this function is called for the first time, the firmware stops any ongoing operation on the device specified and reprograms it according to the channel list supplied. This function uses a fixed CL update frequency – 1953Hz. Thus, the user cannot perform this function call when the layer is involved in any streaming or data mapping operations.

If the user specifies a short timeout delay, this function can time out when called for the first time because it is executed as a pending command and layer programming takes up to 10ms.

Once this function is called, the layer continuously acquires data and every call to the function returns the latest acquired data.

If you would like to cancel ongoing sampling, call the same function with 0xffffffff as a channel number.

Function: **DqAdv211SetCfgChannel**

Syntax: `int DqAdv211SetCfgChannel(int hd, int devn, pDQCFGCH_211 cdata)`

Command: `DQE`

Input:

`int hd`	Handle to the IOM received from DqOpenIOM()
`int devn`	Layer inside the IOM
`pDQCFGCH_211 cdata`	Pointer to 211 channel config structure, see below

Output:

`None`	

Return:

`DQ_NO_MEMORY`	Error allocating buffer
`DQ_ILLEGAL_HANDLE`	Illegal IOM Descriptor or communication wasn't established
`DQ_BAD_DEVN`	Device indicated by devn does not exist or is not an AI-211
`DQ_BAD_PARAMETER`	Illegal value in DQCFGCH_211 structure
`DQ_SEND_ERROR`	Unable to send the Command to IOM
`DQ_TIMEOUT_ERROR`	Nothing is heard from the IOM for timeout duration
`DQ_IOM_ERROR`	Error occurred at the IOM when performing this command
`DQ_SUCCESS`	Successful completion
`Other negative values`	Low level IOM error

Description:

This function sets up the channel configuration parameters for the AI-211. The user may configure as many parameters as he chooses. Any parameter that is not enabled by a <mask> bit will retain its present value.

To accomplish AI-211 channel configuration, the user must allocate, initialize, and pass a pointer to a `pDQCFGCH_211` structure. This structure is defined as:

```
typedef struct {
    uint16 channels;    // channel select bits
```

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

```
    uint16 mask;        // bitwise indicates which of the following struct
                        // fields are valid
    uint16 biasdrive;   // set drive current for iepe sensors, range 0-255
                        // approx = 0-8mA
    uint16 biasonoff;   // DQ_211_BIAS_ON  or  DQ_211_BIAS_OFF
    uint16 comphi;      // 12 bit comparison value for open sensor detection
    uint16 complo;      // 12 bit comparison value for shorted sensor
                        // detection
    uint16 alarmctrl;   // LED alarm control, see defines below
    uint16 hpf;         // high pass filters. see defines below
    uint16 offset;      // test mode.
    uint16 anafilt;     // 48kHz filter. DQ_211_ANALOG_FILTER_ON or
                        // DQ_211_ANALOG_FILTER_OFF
    uint16 main_enb;    // enable dataflow from main ADC, see defines below
    uint16 sec_enbs;    // secondary enables. 0= sec. off, 1= secondary ON
    uint16 secn;        // determine update rate for secondary (LED
                        // comparison)
                       // converter
  } DQCFGCH_211, *pDQCFGCH_211;
```

<channels> flags indicate which channels should be written. The following flags are defined:

```
#define AI211_SEL_CHAN_0        (0x01)
#define AI211_SEL_CHAN_1        (0x02)
#define AI211_SEL_CHAN_2        (0x04)
#define AI211_SEL_CHAN_3        (0x08)
#define AI211_SEL_CHAN_ALL      (0x0f)
```

More than one channel may be specified by or'ing multiple channel flags together.

<mask> flag bits that select which parameters will be written. The following flags are defined:

```
#define DQAI211_CFGCH_DEFAULTSET (1L<<11) // if=1 set all values to default
                                          // state
#define DQAI211_BIASDRIVESET     (1L<<0)  // =1 to set "biasdrive" parameter
#define DQAI211_BIASONOFFSET     (1L<<1)  // =1 to set "biasonoff" parameter
#define DQAI211_COMPHISET        (1L<<2)  // =1 to set "comphi" parameter
#define DQAI211_COMPLOSET        (1L<<3)  // =1 to set "complo" parameter
#define DQAI211_ALARMCTRLSET     (1L<<4)  // =1 to set "alarmctrl" parameter
#define DQAI211_HPFSET           (1L<<5)  // =1 to set "hpf" parameter
#define DQAI211_OFFSETSET        (1L<<6)  // =1 to set "offset" parameter
#define DQAI211_ANAFILTSET       (1L<<7)  // =1 to set "anafilt" parameter
#define DQAI211_MAINENBSET       (1L<<8)  // =1 to set "main_enb" parameter
#define DQAI211_SECENBSSET       (1L<<9)  // =1 to set "secenbs" parameter
#define DQAI211_SECNSET          (1L<<10) // =1 to set "secn" parameter
```

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

The DQAI211_CFGCH_DEFAULTSET flag bit is used to easily set all of the channel configuration values to their default state. Before setting up a custom configuration, it is recommended you first set all channels to their default state by setting <channels> to AI211_SEL_CHAN_ALL, setting <mask> to DQAI211_CFGCH_DEFAULTSET and calling DqAdv211SetCfgChannel().

<biasdrive> sets the amount of drive current for the iepe sensor. When the DQAI211_BIASDRIVESET flag bit is set in <mask>, the <biasdrive> value is set. The following scaling macro is defined to allow for easy translation to the correct values.

```
#define DQ211_DRIVE_CURRENT(I) ((I/8.0)*255) //drive current scaling macro.
```

The driver accepts values from zero to 255, which map to drive currents from zero to 8mA. Values less than zero or greater than 8.0 will return an error.

<biasonoff> Turns the drive current for the iepe sensor on or off . When the DQAI211_BIASDRIVESET flag bit is set in <mask>, the value in <biasonoff> is set. The following two values are defined for this purpose:

```
#define DQ_211_BIAS_ON          (1)
#define DQ_211_BIAS_OFF         (0)
```

<comphi> sets the 12-bit comparison value for the open sensor detector. When the DQAI211_COMPHISET flag bit is set in <mask>, the value in <comphi> is set. Two defines are provided:

```
#define DQ_211_COMP_HI_STD     (0xfa0)  // standard value for doing
                                        // comparison
#define DQ_211_COMP_HI_DEFAULT (0xfff)  // default value, disables comparison
                               // It is suggested you keep the values between
                               // 0xf00 and 0xff0. Setting a value of 0xfff
                               // will turn the comparison off.
```

<complo> sets the 12-bit comparison value for the shorted sensor detector. When the DQAI211_COMPLOSET flag bit is set in <mask>, the value in <complo> is set. Two defines are provided:

```
#define DQ_211_COMP_LO_STD     (0xc0) // standard value for doing comparison
#define DQ_211_COMP_LO_DEFAULT (0x0)  // default value, disables comparison
```

It is suggested you keep the values between 0x30 and 0xe0. Setting a value of zero will turn the comparison off.

<alarmctrl> sets the control settings for the visual alarm LED. When the DQAI211_ALARMCTRLSET flag bit is set in <mask>, the value in <alarmctrl> is set. The following defines are provided:

```
#define DQ_211_ALARM_ON      (0x3)  // LEDs are controlled by comparison
                                    // registers and secondary adc
#define DQ_211_ALARM_OFF     (0)    // LEDs are off
#define DQ_211_ALARM_RED     (0x4)  // LED controlled by program, red on
#define DQ_211_ALARM_GREEN   (0x8)  // LED controlled by program, green on
#define DQ_211_ALARM_ORANGE  (0x0c) // LED controlled by program, orange on
```

<hpf> sets the control settings for high pass filtering or DC coupling. When the DQAI211_HPFSET flag bit is set in <mask>, the value in <hpf> is set. The following defines are provided:

```
#define DQ_211_HPF_DC        (1L<<0)     // DC coupling
#define DQ_211_HPF_POINT1_HZ (1L<<1)     // 0.1 Hz cutoff high pass filter
```

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

```
#define DQ_211_HPF_1_HZ      (1L<<2)     // 1.0 Hz cutoff high pass filter
#define DQ_211_HPF_10_HZ     (1L<<3)     // 10 Hz cutoff high pass filter
```

<offset> sets the control settings for a test mode. This is not normally set by a user. When the DQAI211_OFFSETSET flag bit is set in <mask>, the value in <offset> is set. The following defines are provided:

```
#define DQ_211_OFFSET_TEST_ON   (1)       // test mode on
#define DQ_211_OFFSET_TEST_OFF  (0)       // test mode off - default value
```

When the combination of DQ_211_BIAS_OFF and DQ_211_OFFSET_TEST_ON occurs, the system internally grounds the sensor input for adjustment purposes. No sensor connections are allowed at this time.

<anafilt> sets the 48KHz analog filter ON or OFF. When the DQAI211_ANAFILTSET flag bit is set in <mask>, the value in <anafilt> is set. The following defines are provided:

```
#define DQ_211_ANALOG_FILTER_ON   (1)
#define DQ_211_ANALOG_FILTER_OFF  (0)
```

<main_enb> provides additional control over the main A/D converter for special applications. Control for the main converter is normally provided automatically by the DqAdv211Read() or the ACB and DMap control functions. When the DQAI211_MAINENBSET flag bit is set in <mask>, the value in <main_enb> is set. The following defines are provided:

```
#define DQ_211_MAIN_FLOW_OFF    (0)
#define DQ_211_MAIN_FLOW_ON     (1
```

<sec_enbs> provides control over the secondary A/D converter used by the visual alarm LED function. When the DQAI211_SECENBSSET flag bit is set in <mask>, the value in <sec_enbs> is set. The following defines are provided:

```
#define DQ_211_SEC_ENB_OFF    (0)      // SECondary converter OFF
#define DQ_211_SEC_ENB_LED    (0x1)    // SEC converter updates LED
                                       // comparison
```

The converter must be turned on using the DQ_211_SEC_ENB_LED value in order for the visual alarm LED to function.

<secn> This value sets the number of main converter reads per secondary converter read. When the DQAI211_SECNSET flag bit is set in <mask>, the value in <secn> is set. The following defines are provided:

```
#define DQ_211_SEC_N_STD   (100)     // Number of main reads per SECondary
                                      // read
#define DQ_211_SEC_N_OFF   (0)       // Disable secondary data flow
```

The secondary converter's monitoring of the status of the IEPE sensor connection for the setting of the visual alarm LED does not need to occur at the same rate as the primary A/D converter. This setting allows the user to set the rate at which the secondary converter does this monitoring. The secondary data is also transferred across the isolation barrier at this same rate to be made available for some special functions. Setting the value to DQ_211_SEC_N_OFF disables the transfer of this data, but does not inhibit the functioning of the visual alarm LED. The DQ_211_SEC_N_STD define is the recommended setting for this value, but it may be set to any value from zero to 32,767.

Function: **DqAdv211SetCfgLayer**

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Syntax: `int DqAdv211SetCfgLayer(int hd, int devn, pDQCFGLAYER_211 ldata)`

Command: `DQE`

Input:

`int hd`	Handle to the IOM received from `DqOpenIOM()`
`int devn`	Layer inside the IOM
`pDQCFGLAYER_211 ldata`	Pointer to 211 layer config structure, see below

Output:

`None`	

Return:

`DQ_NO_MEMORY`	Error allocating buffer
`DQ_ILLEGAL_HANDLE`	Illegal IOM Descriptor or communication wasn't established
`DQ_BAD_DEVN`	Device indicated by devn does not exist or is not an AI-211
`DQ_BAD_PARAMETER`	One of the values in the `pDQCFGLAYER_211` structure is set to an illegal value.
`DQ_SEND_ERROR`	Unable to send the Command to IOM
`DQ_TIMEOUT_ERROR`	Nothing is heard from the IOM for timeout duration
`DQ_IOM_ERROR`	Error occurred at the IOM when performing this command
`DQ_SUCCESS`	Successful completion
`Other negative values`	Low level IOM error

Description:

This function sets up advanced layer configuration parameters for the AI-211. These settings apply to all channels on the layer. In the normal case, these configuration settings are set automatically by the `DqAcbInitOps()` function. Use these settings to override standard behavior for special applications. The user may configure as many parameters as he chooses. Any parameter that is not enabled by a `<mask>` bit will retain its present value.

To accomplish AI-211 layer configuration, the user must allocate, initialize, and pass a pointer to a `pDQCFGLAYER_211` structure. This structure is defined as:

```
typedef struct {
    uint16 mask;    // bitwise indicate which of the following fields are
valid
    uint16 clksrc;  // select clock source for divider.
    uint16 clkdiv;  // clock divider. output (1MHz max)
                    // clkdiv=0 passes clksrc w/o change.
                  // Do not set values less than 65 or 23 when 66MHz or 24MHz
                    //  selected.
                    // Sample rate is:  clksrc/((clkdiv+1)*8).
    uint16 fmtr;    // reduced precision data format 1= reduced, 0 = normal.
    uint16 avg_factor; //Set the averaging factor. 0=1, 1=2, 2=4, 3=8, etc.
   } DQCFGLAYER_211, *pDQCFGLAYER_211;
```

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

<mask> flag bits that select which parameters will be written. The following flags are defined:

```
#define DQAI211_CLKSRCSET    (1L<<0) // =1 if "clksrc" contains valid data
#define DQAI211_CLKDIVSET    (1L<<1) // =1 if "clkdiv" contains valid data
#define DQAI211_FMTRSET      (1L<<2) // =1 if "fmtr" contains valid data
#define DQAI211_AVGFACTORSET (1L<<3) // =1 if "avg_factor" contains valid
                                     // data
#define DQAI211_FIR_BY_AVG_SET (1L<<5) // Set default FIR to follow averaging
#define DQAI211_CFGLAYER_DEFAULTSET (1L<<4)  // =1  to set default state
```

The DQAI211_CFGLAYER_DEFAULTSET flag bit is used to easily set all of the layer configuration values to their default state. Before setting up a custom configuration, it is recommended you first set all values to their default state by setting <mask> to DQAI211_CFGLAYER_DEFAULTSET and calling DqAdv211SetCfgLayer(). When DQAI211_CFGLAYER_DEFAULTSET is set, all other <mask> flag bits are ignored.

<clksrc> This value selects the source of the clock to be used to pace the A/D conversion on the layer. This clock source is routed to the clock divider that is set by <clkdiv> below. When the DQAI211_CLKSRCSET flag bit is set in <mask>, the value in <clksrc> is set. The following defines are provided:

```
#define DQ_211_CLK_66MHZ            (0)
#define DQ_211_CLK_24MHZ            (0x10)
#define DQ_211_CLK_SYNC2            (0x18)
#define DQ_211_CLK_SYNC0_BUS        (0x8)
#define DQ_211_CLK_SYNC1_BUS        (0x9)
#define DQ_211_CLK_SYNC2_BUS        (0xa)
#define DQ_211_CLK_SYNC3_BUS        (0xb)
```

<clkdiv> This value sets the clock divider used to set the rate of the A/D conversion. The maximum allowable frequency to pace the main A/D is 1Mhz.

Do not use values less than 65 when 66MHz is selected or less than 23 when 24MHz is selected. The output frequency is clksrc/(clkdiv+1). Max value is 1023. When the DQAI211_CLKDIVSET flag bit is set in <mask>, the value in <clkdiv> is set. The default value is 65.

<fmtr> This value sets a reduced precision mode, reserved for special applications. When the DQAI211_FMTRSET flag bit is set in <mask>, the value in <fmtr> is set. The following defines are provided:

```
#define DQ_211_FMTR_NORMAL          (0)
#define DQ_211_FMTR_REDUCED         (1)
```

The default value is DQ_211_FMTR_NORMAL.

<avg_factor> This value sets the amount of averaging performed. The number of samples averaged together is always a power of 2. Setting <avg_factor> to zero gives no averaging. Setting <avg_factor> to 1 gives 2 samples averaged, setting 2 averages 4 samples, 3 averages 8 , etc. The maximum value for <avg_factor> is 15, which averages 32,768 samples. When the DQAI211_AVGFACTORSET flag bit is set in <mask>, the value in <avg_factor> is set. The

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

DQAI211_FIR_BY_AVG_SET flag bit also uses the data in <avg_factor>. If the DQAI211_FIR_BY_AVG_SET bit is set and DQAI211_AVGFACTORSET is also set, the default FIR filter coefficients will be adjusted to provide the correct low-pass filter response with the factor of 2 lower data rates that come with each increment of the averaging value.

Function: **DqAdv211SetFIR**

Syntax: `int DqAdv211SetFIR(int hd, int devn, int channel, int mask, int decrat, int tapsize, double* data)`

Command: `DQE`

Input:

`int hd`	Handle to the IOM received from `DqOpenIOM()`
`int devn`	Layer inside the IOM
`int channel`	Bit field to select channel, see below
`int mask`	Bit field to select which FIR setting functions to perform, see below
`int decrat`	Desired decimation ratio (NULL if not required), see below
`int tapsize`	Number of taps in filter, length of the following *data (NULL if not required), see below
`double *data`	Pointer to filter taps data (NULL if not required), see below

Output:

`None`	

Return:

`DQ_NO_MEMORY`	Error allocating buffer
`DQ_ILLEGAL_HANDLE`	Illegal IOM Descriptor or communication wasn't established
`DQ_BAD_DEVN`	Device indicated by devn does not exist or is not an AI-211
`DQ_BAD_PARAMETER`	No channel specified, decimation ratio is illegal value, tapsize is illegal value or *data is NULL
`DQ_SEND_ERROR`	Unable to send the Command to IOM
`DQ_TIMEOUT_ERROR`	Nothing is heard from the IOM for timeout duration
`DQ_IOM_ERROR`	Error occurred at the IOM when performing this command
`DQ_SUCCESS`	Successful completion
`Other negative values`	Low level IOM error

Description:

This function can be used to perform all FIR configuration and control functions for the AI-211. In the normal case, these configuration settings are set automatically by the `DqAcbInitOps()` function. Use these settings to override standard behavior for special applications. The user may configure as many parameters as he chooses. Any parameter that is not enabled by a <`mask`> bit will retain its present value.

<`channel`> parameter indicates which channels will get their FIR configuration changed. The following flags are defined:

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

```
#define AI211_SEL_CHAN_0        (0x01)
#define AI211_SEL_CHAN_1        (0x02)
#define AI211_SEL_CHAN_2        (0x04)
#define AI211_SEL_CHAN_3        (0x08)
#define AI211_SEL_CHAN_ALL      (0x0f)
```

More than one channel may be specified by or'ing multiple channel flags together.

<mask> parameter flag bits that select which parameters will be written. The following flags are defined:

```
#define AI211_FIR_SET_DEFAULT         (0x8) //set and enable the default
                                           //filter
#define AI211_FIR_COEFF_LOAD          (0x4) // load taps and coefficients
#define AI211_FIR_SET_DECIMATION_RATE (0x2) // set decimation rate
#define AI211_FIR_ENABLE              (0x1) // enable fir filter
#define AI211_FIR_DISABLE             (0x0) // disable fir filter
#define AI211_FIRFIRST_ENABLE         (0x0) // perform FIR before averaging
#define AI211_FIRFIRST_DISABLE        (0x10)// perform FIR after averaging
```

The AI211_FIR_SET_DEFAULT flag bit is used to easily set all of the FIR configuration values to their default state. Before setting up a custom configuration, it is recommended you first set all channels to their default state by setting <channel> to AI211_SEL_CHAN_ALL, setting <mask> to AI211_FIR_SET_DEFAULT and calling DqAdv211SetFIR(). When the AI211_FIR_SET_DEFAULT flag is set, all of the other <mask> parameter bits are ignored.

The AI211_FIR_ENABLE and AI211_FIR_DISABLE defines are used to enable or bypass the FIR filter when a custom FIR filter has been loaded. When standard default settings are used, the system will automatically load and enable the FIR filter. No additional enabling is required.

The AI211_FIRFIRST_ENABLE and AI211_FIRFIRST_DISABLE defines are used to set the order in which the FIR filtering and the averaging takes place. When standard default settings are used, the system will automatically set the FIR filtering to occur first.

<decrat> parameter sets the decimation rate. Zero = no decimation. Setting a value of 1 discards one reading for every one kept, two discards two readings for every one kept, etc. When the AI211_FIR_SET_DECIMATION_RATE flag bit is set in <mask> parameter, the value in <decrat> is set.

The default value is zero.

<tapsize> and <*data> parameters are used to load user-defined filter coefficients to the selected FIR filters. <tapsize> is the number of coefficients expected in the array of double precision coefficients pointed to by <*data>. The coefficients should be in the [-1.0…1.0] range. The sum of all the taps should equal 1.0. When the AI211_FIR_COEFF_LOAD flag bit is set in the <mask> parameter, the <tapsize> and <*data> values are set.

Function: **DqAdv211SetPll**

Syntax: `int DqAdv211SetPll(int hd, int devn, double samplerate, double* sr_actual, int* avg_fact, int line)`

Command: DQE

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Input:

`int hd`	Handle to the IOM received from `DqOpenIOM()`
`double samplerate`	Desired sampling rate
`int line`	Sync line to assign signal

Output:

`double* sr_actual`	Actual sample rate
`int *avg_fact`	Required averaging factor for AI-211 layer

Return:

`DQ_BAD_PARAMETER`	Requested sample rate is too high or too low.
`DQ_SEND_ERROR`	Unable to send the Command to IOM
`DQ_TIMEOUT_ERROR`	Nothing is heard from the IOM for timeout duration
`DQ_IOM_ERROR`	Error occurred at the IOM when performing this command
`DQ_SUCCESS`	Successful completion
`Other negative values`	Low level IOM error

Description:

The PLL clock circuit on the CPU layer will be programmed and routed by this function. The PLL may be routed by way of the sync[1] or sync[3] lines by using the `DQ_EXT_SYNC1` or `DQ_EXTSYNC3` constants for the “line” parameter.

The AI-211 uses clock rates that are higher than the sampling rate and also uses data averaging to increase resolution. The function will calculate the correct clock and averaging values for the AI-211. The returned average factor should be sent to the AI-211 layers using the `DqAdv211SetCfgLayer()` function as follows:

```
 DQCFGLAYER_211 ldata;
ldata.mask = (DQAI211-AVGFACTORSET | DQAI211_FIR_BY_AVG_SET);
LDATA.AVG_FACTOR = (uint16)average_factor;
DqAdv211SetCfgLayer (hd0,DEVN,&ldata);

The Config word used by the second parameter of DqAcbInitOps() must
enable the PLL clock by specifying the DQ_LN_CVCKSRC1 constant. For
example:
#define CFG211      (DQ_ENABLED   \
                    |DQ_LN_ACTIVE \
                    |DQ_LN_GETRAW \
                    |DQ_LN_IRQEN   \
                    |DQ_LN_CVCKSRC1 \
                    |DQ_LN_STREAMING \
                    |DQ_FIFO_MODEFIFO)
```

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Appendix A

Accessories

A.1 DNA-ACC-211 Breakout Board

The DNA-ACC-211 Breakout Board is an easy-to-use, versatile, accessory for direct connection of piezoelectric and other 2-wire vibration sensors to the DNx-AI-211 board. It can accept signals from 4 ICP, IEPE, and other vibration sensors.

Since the Breakout Board is supplied with a DB-37 board-mounted connector that mates directly with the I/O connector on a DNA-AI-211 or DNR-AI-211 board, it can be plugged directly into the DB-37 connector of the 211 board. A photo of the device is shown in **Figure A-1** below.

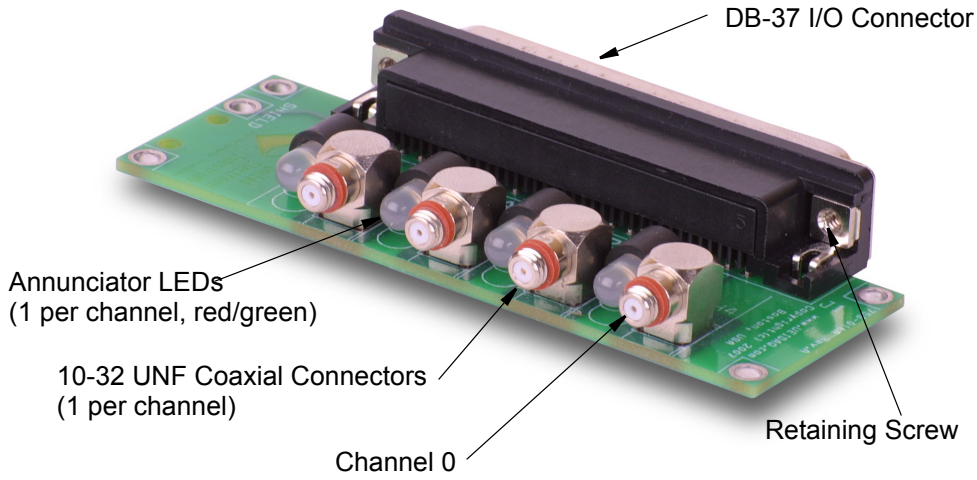


Figure A-1 Photo of DNA-ACC-211 Breakout Board

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

The Technical Specifications for the DNA-ACC-211 are listed in the table below.

Table 4-1. DNA-ACC-211 Specifications

Item	Specification
No. of channels	4
Connector Type	Standard 10-32 UNF coaxial connectors
Annunciator LEDs States: Green Solid Red Flashing Red Off Orange (red + green)	1 per channel, red/green, next to UNF connectors Indicates: OK Open Sensor Shorted Sensor No bias current Not a valid state.
ESD Protection	15 kV
Operating Temperature	Tested -40°C to +85°C
Vibration *IEC 60068-2-6* *IEC 60068-2-64*	5 g, 10-500 Hz, sinusoidal 5 g (rms), 10-500 Hz, broadband random
Shock *IEC 60068-2-27*	50 g, 3 ms half sine 18 shocks @ 6 orientations 30 g, 11 ms half sine 18 shocks @ 6 orientations
Humidity	0 to 95%, non-condensing
Altitude	0 to 70,000 feet
Isolation	350 VDC (chan-to-chan, chan-to-chassis)

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

4.1.1 Mounting Multiple DNA-ACC-211 Boards

If your installation uses multiple DNA-AI-211 boards inserted into a DNA Cube in adjacent slot positions, you must use special care and hardware (supplied with the units) in mounting the associated ACC-211 boards so as to avoid short circuits between coaxial connector on adjacent boards. To do this, refer to **Figure A-2** and execute the procedure described below.

NOTE: This procedure (and hardware) are not required with the DNR-AI-211 mounted in a RACKtangle enclosure because spacing between board positions is greater.

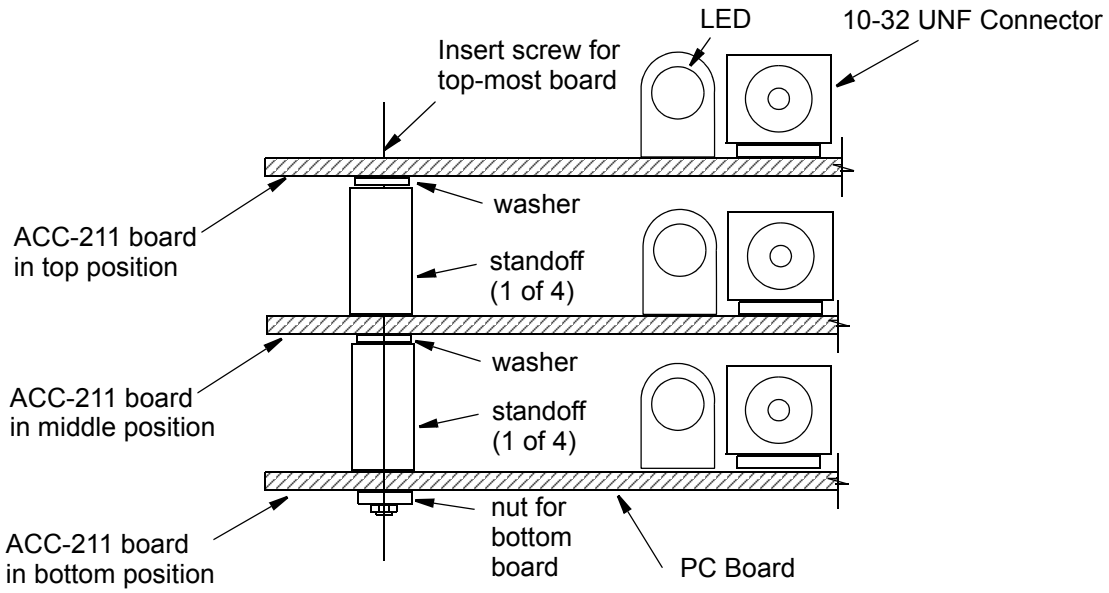


Figure A-2 Mounting Multiple DNA-ACC-211 Breakout Boards

STEP 1: If multiple ACC-211 breakout boards are being used, pre-assemble them before attaching them to the DNA-AI-211 boards in the Cube.

STEP 2: Pre-assemble the multiple ACC-211 boards by mounting standoffs, washers, nuts, and retaining screws at each corner of the board as shown in **Figure A-2**. Be sure to use a screw in the top-most beakout board and a nut for the bottom board.

STEP 3: When pre-assembly is complete, carefully plug the assembled breakout boards into the DB-37 I/O Connectors on the DNA-AI-211 boards mounted in the Cube and tighten the retaining screws on the I/O connectors.

STEP 4: Inspect the final assembly to verify that no shorts exist between the 10-32 UNF coaxial connectors on adjacent boards.

A.2 Other Accessories

In addition to the DNA-ACC-211 breakout board, the following cable is available for the AI-211 layer.

- **DNA-CBL-BNC Cable**
  2-foot coaxial cable with a 10-32 UNF male connector on one end and a full-size male BNC Cable/Adaptor connector on the other end. Use for connecting a piezoelectric vibration sensor to the DNA-ACC-211 breakout board.

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

A.3 Layer Calibration

Please note that once you perform layer calibration yourself, the factory calibration warranty is void. For AI-211 layers, we recommend annual factory recalibration at UEI.

© Copyright 2009 all rights reserved
United Electronic Industries, Inc.

Index

A
Annunciator LEDs 8
Antialiasing FIlters 7
Architecture 5

B
Block Diagram 6

C
Calibration 27
Connector, DB-37 8
Connectors 8

D
Digital Filter 7
DNA-ACC-211 Breakout Board 8
DNA-ACC-211 Connector Pinout 9
DNA-CBL-37 Cable 26

H
High-Level API 11

J
Jumper Settings 9

L
Layer Position Jumper Settings 10

M
Manual Conventions 2
Manual Organization 1
Measuring Circuit for a Vibration Sensor 5

P
Photo of DNA-ACC-211 Breakout Board 24
photo of DNA-AI-211 Board 4

S
Specifications of DNA-ACC-211 25

© Copyright 2009
United Electronic Industries, Inc.